

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Коваль Никита Сергеевич

Магистерская диссертация

**Многоцелевое управление процессом
динамического позиционирования морских судов**

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа Прикладная математика и информатика в задачах
цифрового управления

Научный руководитель,
доктор физ.-мат. наук,
профессор
Веремей Е.И.

Санкт-Петербург

2018

Оглавление

Введение.....	3
Постановка задачи	5
Обзор литературы	7
Глава 1. Многоцелевое управление с прогнозом.....	9
Глава 2. Оптимизация функционала качества в задаче управления с прогнозом.....	13
2.1. Дифференцируемость целевого функционала.....	13
2.2. Выбор метода оптимизации.....	17
Глава 3. Математическая модель морского судна.....	19
Глава 4. Имитационное моделирование движения морского судна.....	21
4.1. Компьютерная модель морского судна	21
4.2. Сравнение методов вычисления градиента.....	23
4.3. Анализ качества функционирования системы управления	24
Выводы.....	27
Заключение	28
Список литературы	29
Приложение	31

Введение

Система динамического позиционирования морского судна – это комплекс аппаратного и программного обеспечения, который обеспечивает перевод судна в желаемое положение на горизонтальной плоскости с последующей стабилизацией. Она включает в себя ряд компонентов:

- информационно-измерительные системы;
- движительно-рулевой комплекс;
- центральный бортовой компьютер.

Важным требованием к системе динамического позиционирования является возможность удержания судна в желаемом положении посредством лишь движительно-рулевого комплекса. Этот комплекс обычно включает в себя стабилизирующие двигатели, подруливающие устройства, рулевые колонки Арнессона и т.д.

Традиционный метод стабилизации судна заключается в использовании якорных систем позиционирования. Однако этот способ имеет ряд недостатков. Во-первых, этот вариант может использоваться только в неглубоких водоемах (до 500 метров). Во-вторых, в этом случае смена желаемого положения является весьма трудозатратой операцией. Системы динамического позиционирования лишены этих недостатков.

Удержание судна в желаемом положении осложняется различными возмущениями, такими как ветер, морское волнение, течения, которые действуют на судно. Система динамического позиционирования должна успешно идентифицировать эти возмущения и производить их компенсацию, иначе это приведет к существенному отклонению судна от желаемого положения. В результате могут возникать сбои или даже аварии во время проведения работ. Это, в свою очередь влечет за собой серьезные экономические и экологические проблемы.

Существует множество направлений, которые требуют поддержки

судном динамического позиционирования. Это разведка и добыча природных ресурсов, обеспечение водолазных работ, поддержка процессов возведения и ремонта элементов подводной инфраструктуры, научно-исследовательские работы и множество других приложений.

Постановка задачи

Как было отмечено во введении, задача динамического позиционирования заключается в удержании морского судна в определенной точке на горизонтальной плоскости с заданным углом курса. Важно, что стабилизация должна обеспечиваться исключительно посредством движительно-рулевого комплекса системы.

В качестве оценки положения судна чаще всего выступает сервис DGPS. Основная идея функционирования этой системы заключается в использовании дополнительного приемника на берегу с известной позицией, который используется для вычисления погрешности GPS. При таком подходе точность измерения координат судна доходит до 1 метра. Для измерения курса судна обычно используется гирокомпас, который имеет погрешность порядка $0,1^\circ$. Измерение скорости является технически более сложной задачей и не всегда осуществляется в информационно-измерительных системах. Поэтому, в данной задаче будем считать скорость недоступной для измерения.

Так как реализация алгоритма будет проводиться на цифровом компьютере, следует рассматривать дискретные законы управления. Кроме того, современные устройства обладают мощностью способной в реальном времени решать трудоемкие задачи. В связи с этим, в настоящее время популярность приобретает алгоритм управления с прогнозом. Главная идея этого метода заключается в моделировании движения судна на некотором горизонте прогноза и выборе оптимального программного управления.

Оптимизация управляющего сигнала проводится в соответствии с некоторым функционалом, который обычно формируется с учетом нескольких характеристик. Чаще всего это степень отклонения системы от желаемого положения и величина управления необходимая для решения задачи. Закон управления, производящий оптимизацию по нескольким

параметрам системы, называется многоцелевым.

Алгоритм управления с прогнозом имеет ряд достоинств. Во-первых, это относительная простота формирования схемы обратной связи. Во-вторых, такой закон управления оказывается весьма адаптивным к различным особенностям системы. И наконец, выбранное управление является оптимальным для заданного горизонта прогноза.

Целью данной работы является применение алгоритма управления с прогнозом для решения задачи динамического позиционирования морского судна, а так же поиск эффективных способов реализации этого подхода.

Обзор литературы

История развития систем динамического позиционирования морских судов начинается в 60-е годы. Первые алгоритмы были основаны на PID-регуляторах с использованием фильтра относительно высоких частот для фильтрации волнения. В 70-е годы появляются системы, построенные на основе фильтра Калмана. Примеры таких систем описаны в работах [1] (Balchen, Jenssen, и Saelid) и 1983 [3] (Fung and Grimble). Применение этих систем предполагало линеаризацию уравнений движения для заданных углов курса, определяемых разбиением отрезка $[0, 360^\circ]$ на равные части. Такой способ построения закона управления был весьма громоздким и требовал настройки большого числа параметров. Кроме того, этот подход обеспечивал лишь локальную устойчивость системы.

Далее, в 90-е интерес сместился в сторону нелинейных систем для преодоления указанных недостатков. Например, в Fossen и Strand [2], Strand и Fossen [10] и Strand [9], Loria, Fossen, Panteley [6] описываются законы управления, основанные на нелинейном наблюдателе, который оценивает внешние возмущения и производит фильтрацию волнения для вектора состояния судна. В статье Е. Veremey [12] в закон управления дополнительно введен динамический корректор.

В настоящее время ведутся исследования в области оптимальных систем динамического позиционирования. Наиболее распространенным подходом, реализующим эту концепцию, является управление с прогнозом. Однако существует ряд трудностей при применении этого подхода. В особенности это касается нелинейного варианта алгоритма. В этом случае требуется оптимизация нелинейного функционала, зависящего от большого числа аргументов, что, во-первых, является само по себе трудоемкой задачей, во-вторых требует специальных методов для гарантии нахождения глобального минимума.

В работе [5] дано обозрение общих принципов построения нелинейных систем динамического позиционирования с прогнозом. Предлагается использование монолитной архитектуры формирования управляющего сигнала. Другой вариант алгоритма предложен в [11]. Здесь закон управления производит переключение между линейным и нелинейным алгоритмом управления в зависимости от величины отклонения судна от желаемого положения. Подобный подход можно считать весьма эффективным в смысле качества процессов управления, однако у него есть существенный недостаток, заключающийся в относительно низкой надежности.

Глава 1. Многоцелевое управление с прогнозом

Рассмотрим общую постановку дискретной задачи управления с прогнозом. В этом случае управление с обратной связью формируется именно на основе оптимизации прогнозируемого движения.

Будем полагать, что измерение положения объекта и задание управляющего сигнала происходит в фиксированные, равноотстоящие моменты времени $t_k = t_0 + kT_s$, где T_s период дискретизации, k – номер отсчета. Таким образом, управляющий сигнал – это кусочно-постоянная функция, сохраняющая свое значение на сегментах $\Delta_k = [t_k, t_{k+1})$.

Пусть нам известны дифференциальные уравнения движения объекта:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\tau}, \mathbf{b}), \quad (1)$$

где $\mathbf{x} \in \mathbf{R}^n$ – вектор состояния, $\boldsymbol{\tau} \in \mathbf{U} \subseteq \mathbf{R}^r$ – управляющий вектор, $\mathbf{b} \in \mathbf{R}^r$ – вектор внешних возмущений, $\mathbf{f} \in \mathbf{R}^n$ – функция, задающая движение \mathbf{R}^n объекта.

Введем также уравнения измерения:

$$\boldsymbol{\eta} = \mathbf{C}\mathbf{x}.$$

Если поставить задачу Коши $\mathbf{x}(t_0) = \mathbf{x}_0$ для уравнения (1), а также определить управление и возмущение на промежутке $[t_0, t]$, то решение будет иметь вид:

$$\mathbf{x} = \boldsymbol{\varphi}(t, t_0, \mathbf{x}_0, \boldsymbol{\tau}(\cdot), \mathbf{b}(\cdot)). \quad (2)$$

Проведем дискретизацию непрерывной системы. Для этого обозначим через $\mathbf{x}[k]$ вектор состояния объекта в момент времени t_k :

$$\mathbf{x}[k] = \boldsymbol{\varphi}(t_k, t_0, \mathbf{x}_0, \boldsymbol{\tau}([t_0, t_k]), \mathbf{b}([t_0, t_k])).$$

В силу свойств решений дифференциальных уравнений можно

записать следующее равенство:

$$\mathbf{x}[k+1] = \boldsymbol{\varphi}(t_{k+1}, t_k, \mathbf{x}[k], \boldsymbol{\tau}([t_k, t_{k+1}]), \mathbf{b}([t_k, t_{k+1}])),$$

где $\boldsymbol{\tau}([t_k, t_{k+1}])$, $\mathbf{b}([t_k, t_{k+1}])$, – соответственно управляющий и возмущающий сигналы на промежутке $[t_k, t_{k+1}]$. Так как на этом промежутке функция $\boldsymbol{\tau}$ не меняется, то её можно заменить значением на этом промежутке, которое обозначим $\boldsymbol{\tau}[k]$. Положим, что возмущения \mathbf{b} также постоянны на этом промежутке и принимают значение $\mathbf{b}[k] = \mathbf{b}(t_k)$. Получим уравнение следующего вида:

$$\mathbf{x}[k+1] = \boldsymbol{\varphi}(t_{k+1}, t_k, \mathbf{x}[k], \boldsymbol{\tau}[k], \mathbf{b}[k]).$$

В итоге получаем разностное уравнение:

$$\mathbf{x}[k+1] = \mathbf{F}(\mathbf{x}[k], \boldsymbol{\tau}[k], \mathbf{b}[k]). \quad (3)$$

Правая часть в этом уравнении имеет вид:

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\tau}, \mathbf{b}) = \boldsymbol{\varphi}(t_{k+1}, t_k, \mathbf{x}, \boldsymbol{\tau}, \mathbf{b}).$$

Наконец, обозначим $\boldsymbol{\eta}[k]$ измеряемые переменные объекта, которые связаны с вектором состояния следующим уравнением:

$$\boldsymbol{\eta}[k] = \mathbf{C}\mathbf{x}[k].$$

Пусть в момент времени k объект находится в состоянии $\mathbf{x}[k]$ и на него действует вектор возмущений $\mathbf{b}[k]$. Необходимо выбрать управляющий сигнал $\boldsymbol{\tau}[k]$ для перехода в новое состояние.

Зададим управление $\boldsymbol{\tau}[k+i]$ для $j=1..C$. Так как на промежутке $[t_k, t_{k+P-1}]$ возмущения не известны, следует положить их равными $\mathbf{b}[k]$. Вообще говоря, величины $\mathbf{x}[k]$ и $\mathbf{b}[k]$ не доступны для измерения. Однако, существуют методы их оценки. Чаще всего для этих целей используют асимптотические

наблюдатели.

Константы P и C называются горизонтом прогноза и горизонтом управления, соответственно. Если $C < P$, то остальные отсчеты управления можно положить либо нулями, либо равными вектору $\tau[k+C-1]$.

Рассмотрим движение объекта $x[k+j]$ на промежутке $j=1..P$ и введем следующие обозначения:

$$\bar{\eta} = (\eta[k+1]^T, \dots, \eta[k+P]^T)^T,$$

$$\bar{\tau} = (\tau[k]^T, \dots, \tau[k+P-1]^T)^T.$$

Введем функционал качества на полученном движении следующим образом:

$$J(\bar{\eta}, \bar{\tau}) = \sum_{i=1}^P J_i(\eta[k+i], \tau[k+i-1]) =$$

$$= \sum_{i=1}^P (\eta[k+i] - \eta_d)^T Q_\eta (\eta[k+i] - \eta_d) + \tau[k+i-1]^T Q_\tau \tau[k+i-1], \quad (4)$$

где Q_η, Q_τ – симметрические положительно определенные матрицы.

В данном функционале первое слагаемое под знаком суммы отвечает за точность стабилизации объекта в желаемом положении, а второе – позволяет регулировать интенсивность функционирования управления, которая используется для стабилизации.

Заметим, что последовательность векторов состояния зависит от управления, поэтому значение функционала определяется выбором вектора $\bar{\tau}$:

$$J(\bar{\eta}(\bar{\tau}), \bar{\tau}) = J(\bar{\tau}).$$

Поставим следующую оптимизационную задачу:

$$\tau^* = \arg \min_{\bar{\tau} \in U} J(\bar{\tau}).$$

Решение τ^* этой задачи является оптимальным программным

управлением на промежутке $[t_k, t_{k+P}]$.

Теперь в качестве управляющего сигнала $\tau[k]$ возьмем вектор, который состоит из первых m компонент вектора τ^* .

Таким образом, алгоритм управления с прогнозом заключается в решении задачи минимизации функции $C \cdot m$ переменных. Вычисление значения этого функционала для выбранной последовательности управляющих сигналов требует решения дифференциального уравнения (1) на промежутке $[t_0, t_k]$.

Глава 2. Оптимизация функционала качества в задаче управления с прогнозом

Устойчивость алгоритма управления с прогнозом является темой отдельного исследования. Однако во многом она зависит от величины горизонта прогноза. При этом, увеличение горизонта влечет за собой усложнение задачи оптимизации. Таким образом, требуются эффективные способы поиска минимума функционала. Это требует выбора подходящего численного метода.

В этой главе будут выведены условия дифференцируемости функционала, предложен алгоритм вычисления градиента, а также приведены соображения по поводу выбора подходящего метода оптимизации.

2.1. Дифференцируемость целевого функционала

В этом параграфе покажем, что при выполнении некоторых условий для функции правых частей дифференциального уравнения (1) функционал (4) дифференцируем и затем рассмотрим алгоритм эффективного вычисления его градиента.

Перед этим приведем условия дифференцируемости функции F правых частей разностных уравнений объекта по переменным x и τ . С этой целью рассмотрим решение (2) дифференциального уравнения (1) на некотором промежутке $[t', t'']$:

$$x = \varphi(t, t', \tilde{x}, \tilde{\tau}, \tilde{b}).$$

Предположим, что на этом промежутке вектора управления и возмущения не меняется. Таким образом, они могут выступать в качестве числовых параметров. Если функции f , $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial \tau}$ – непрерывны, то решение этого уравнения $\varphi(t, t', \tilde{x}, \tilde{\tau}, \tilde{b})$ дифференцируемо как по $\tilde{\tau}$, так и по \tilde{x} [15].

Введем следующие обозначения:

$$\mathbf{A}(t) = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \tilde{\mathbf{\tau}}, \tilde{\mathbf{b}})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\boldsymbol{\varphi}(t, t', \tilde{\mathbf{x}}, \tilde{\mathbf{\tau}}, \tilde{\mathbf{b}})},$$

$$\mathbf{B}(t) = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \tilde{\mathbf{\tau}}, \tilde{\mathbf{b}})}{\partial \tilde{\mathbf{\tau}}} \right|_{\mathbf{x}=\boldsymbol{\varphi}(t, t', \tilde{\mathbf{x}}, \tilde{\mathbf{\tau}}, \tilde{\mathbf{b}})}.$$

Тогда матрицы Якоби $\mathbf{X}(t) = \frac{\partial \boldsymbol{\varphi}(t, t', \tilde{\mathbf{x}}, \tilde{\mathbf{\tau}}, \tilde{\mathbf{b}})}{\partial \tilde{\mathbf{x}}}$ и $\mathbf{Y}(t) = \frac{\partial \boldsymbol{\varphi}(t, t', \tilde{\mathbf{x}}, \tilde{\mathbf{\tau}}, \tilde{\mathbf{b}})}{\partial \tilde{\mathbf{\tau}}}$

удовлетворяют уравнениям в вариациях:

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{A}(t)\mathbf{X}(t), \\ \mathbf{X}(t') &= \mathbf{E}_n, \end{aligned} \tag{5}$$

$$\begin{aligned} \dot{\mathbf{Y}}(t) &= \mathbf{A}(t)\mathbf{Y}(t) + \mathbf{B}(t), \\ \mathbf{Y}(t') &= \mathbf{O}_{n \times m}. \end{aligned} \tag{6}$$

Решение этих уравнений представимо в следующем виде:

$$\mathbf{X}(t) = \boldsymbol{\Phi}(t)\boldsymbol{\Phi}(t')^{-1},$$

$$\mathbf{Y}(t) = \int_{t'}^t \boldsymbol{\Phi}(t)\boldsymbol{\Phi}(s)^{-1}\mathbf{B}(s)ds,$$

где $\boldsymbol{\Phi}(t)$ – фундаментальная матрица системы (5).

Теперь, для того чтобы вычислить производные функции \mathbf{F} для некоторого отсчета l , можно использовать следующие формулы:

$$\frac{\partial \mathbf{F}(\mathbf{x}, \boldsymbol{\tau}, \mathbf{b})}{\partial \mathbf{x}} = \boldsymbol{\Phi}(t_{l+1})\boldsymbol{\Phi}(t_l)^{-1}, \tag{7}$$

$$\frac{\partial \mathbf{F}(\mathbf{x}, \boldsymbol{\tau}, \mathbf{b})}{\partial \boldsymbol{\tau}} = \int_{t_l}^{t_{l+1}} \boldsymbol{\Phi}(t_{l+1})\boldsymbol{\Phi}(s)^{-1}\mathbf{B}(s)ds. \tag{8}$$

Прейдем к поиску градиента целевого функционала. Отметим, что приведенный алгоритм основан на идеях, изложенных в книге [8]. Не умаляя общности, положим $k = 0$. Найдем градиент отдельного слагаемого $J_i(\boldsymbol{\tau})$. Как можно заметить, оно зависит только от переменных $\boldsymbol{\tau}[1], \dots, \boldsymbol{\tau}[i]$, если $i \leq C$.

Обозначим $i_C = \min(i, C)$.

Найдем его градиент по $\tau[j], j \leq i$.

$$\begin{aligned} \frac{\partial J_i}{\partial \tau[j]} &= \frac{\partial}{\partial \tau[j]} \left[(\eta[i] - \eta_d)^T Q_\eta (\eta[i] - \eta_d) + \tau[i]^T Q_\tau \tau[i] \right] = \\ &= 2(\eta[i] - \eta_d)^T Q_\eta C \frac{\partial \mathbf{x}[i]}{\partial \tau[j]} + 2\tau[j]^T Q_\tau \delta_{ij}, \end{aligned} \quad (9)$$

где δ_{ij} символ Кронекера.

Так как $\mathbf{x}[i]$ является решением разностного уравнения (3), то справедливы следующие выкладки:

$$\begin{aligned} \frac{\partial \mathbf{x}[i]}{\partial \tau[j]} &= \frac{\partial \mathbf{F}(\mathbf{x}[i-1], \tau[i-1], \mathbf{b}[i-1])}{\partial \mathbf{x}[i-1]} \frac{\partial \mathbf{x}[i-1]}{\partial \tau[j]} = \\ &= \frac{\partial \mathbf{F}(\mathbf{x}[i-1], \tau[i-1], \mathbf{b}[i-1])}{\partial \mathbf{x}[i-1]} \frac{\partial \mathbf{F}(\mathbf{x}[i-2], \tau[i-2], \mathbf{b}[i-2])}{\partial \mathbf{x}[i-2]} \frac{\partial \mathbf{x}[i-2]}{\partial \tau[j]} = \dots = \\ &= \frac{\partial \mathbf{F}(\mathbf{x}[i-1], \tau[i-1], \mathbf{b}[i-1])}{\partial \mathbf{x}[i-1]} \dots \frac{\partial \mathbf{F}(\mathbf{x}[j+1], \tau[j+1], \mathbf{b}[j+1])}{\partial \mathbf{x}[j+1]} \frac{\partial \mathbf{F}(\mathbf{x}[j], \tau[j], \mathbf{b}[j])}{\partial \tau[j]}. \end{aligned}$$

Так как матрица \mathbf{A} непрерывна на промежутке $[t_0, t_P]$, фундаментальная матрица так же будет определена на всем этом промежутке. Применим формулы (7), (8) и выберем фундаментальную матрицу общей для всех якобианов:

$$\begin{aligned} \frac{\partial \mathbf{x}[i]}{\partial \tau[j]} &= \Phi(t_i) \Phi(t_{i-1})^{-1} \dots \Phi(t_{j+1}) \Phi(t_j)^{-1} \int_{t_{j-1}}^{t_j} \Phi(t_j) \Phi(s)^{-1} \mathbf{B}(s) ds = \\ &= \Phi(t_i) \int_{t_{j-1}}^{t_j} \Phi(s)^{-1} \mathbf{B}(s) ds. \end{aligned}$$

Обозначим $\int_{t_{j-1}}^{t_j} \Phi(s)^{-1} \mathbf{B}(s) ds = \Theta_j$. В итоге вычисление якобиана $\frac{\partial \mathbf{x}[i]}{\partial \tau[j]}$

сводится к вычислению фундаментальной матрицы $\Phi(t)$ для $t=t_1, \dots, t_P$ и интеграла Θ_l для $j=1..C$. $\Phi(t)$ можно найти, решая уравнение (5) на промежутке $[t_0, t_P]$ с начальными условиями $\mathbf{X}(t_0) = \mathbf{E}_n$. Чтобы получить

значение интеграла Θ_i , для каждого $i=1..C$ просуммируем матрицы Якоби $\frac{\partial \mathbf{x}[i]}{\partial \boldsymbol{\tau}[j]}$ по $j=1..i$

$$\sum_{j=1}^i \frac{\partial \mathbf{x}[i]}{\partial \boldsymbol{\tau}[j]} = \Phi(t_i) \sum_{j=1}^i \Theta_j = \Phi(t_i) \sum_{j=1}^i \int_{t_{j-1}}^{t_j} \Phi(s)^{-1} \mathbf{B}(s) ds = \Phi(t_i) \int_{t_0}^{t_i} \Phi(s)^{-1} \mathbf{B}(s) ds$$

Заметим, что матрицу $\Phi(t_i) \int_{t_0}^{t_i} \Phi(s)^{-1} \mathbf{B}(s) ds$ можно вычислить, решая уравнение (6) на промежутке $[t_0, t_C]$ с начальными условиями $\mathbf{Y}(t_0) = \mathbf{O}_{n \times m}$. Суммы предыдущего равенства могут быть получены из решения этого уравнения следующим образом:

$$\Phi(t_i) \sum_{j=1}^i \Theta_j = \mathbf{Y}(t_i)$$

Преобразовав это равенства можно получить формулы для вычисления интегралов Θ_i

$$\Theta_1 = \int_{t_0}^{t_1} \Phi(s)^{-1} \mathbf{B}(s) ds = \Phi(t_1)^{-1} \mathbf{Y}(t_1)$$

$$\Theta_j = \int_{t_{j-1}}^{t_j} \Phi(s)^{-1} \mathbf{B}(s) ds = \Phi(t_j)^{-1} \mathbf{Y}(t_j) - \Phi(t_{j-1})^{-1} \mathbf{Y}(t_{j-1})$$

Если учесть, что $\mathbf{G}(t_0) = \mathbf{O}_{n \times m}$, то справедлива следующая формула

$$\frac{\partial \mathbf{x}[i]}{\partial \boldsymbol{\tau}[j]} = \Phi(t_i) \left[\Phi(t_j)^{-1} \mathbf{Y}(t_j) - \Phi(t_{j-1})^{-1} \mathbf{Y}(t_{j-1}) \right] \quad (10)$$

Для того, чтобы получить итоговый градиент, нужно просуммировать найденные градиенты J_i .

Итоговый алгоритм включает в себя четыре шага:

1. численное решение нелинейного уравнения (1) на промежутке $[t_k, t_{k+P}]$;
2. поиск решения матричных уравнений (5), (6) в точках t_k, \dots, t_{k+P} ;

3. вычисление матриц Якоби $\frac{\partial \mathbf{x}[i]}{\partial \mathbf{t}[j]}$ по формуле (10);

4. вычисление градиентов слагаемых J_i по формуле (9) и их суммирование.

Таким образом, градиент функционала может быть эффективно вычислен. Для разностной оценки градиента необходимо получение значений функционала в $C \cdot m + 1$ точке. Одно вычисление функционала требует решения дифференциального уравнения порядка n на промежутке $[t_0, t_P]$, поэтому итоговое время работы будет равно $(C \cdot m + 1) \cdot \text{ode}(P)$, где $\text{ode}(P)$ – время решения дифференциального уравнения для горизонта прогноза P .

Рассмотренный же метод требует лишь решение нелинейного уравнения (1) на промежутке $[t_0, t_P]$, двух матричных уравнений (5) и (6), а также дополнительные манипуляции с матрицами. В совокупности получается асимптотика $O((n+m+1) \cdot \text{ode}(P) + C \cdot P \cdot n^3)$.

Отсюда видно, что в рассмотренном методе при увеличении горизонта управления количество решений дифференциальных уравнений не увеличивается. Кроме того, разностные схемы содержат методические погрешности оценки градиента, в то время, как предложенный способ вычисляет его точное значение.

2.2. Выбор метода оптимизации

В предыдущем параграфе рассмотрены условия, которые должны быть наложены на функцию правых частей уравнения объекта для того, чтобы целевой функционал был дифференцируемым. Здесь же будут приведены некоторые соображения по выбору подходящего метода оптимизации для таких функционалов.

Сразу можно отметить, что дифференцируемость функционала (4)

позволяет использовать методы первого порядка. Однако применение градиентных методов может быть неэффективным в данном случае. Функционал имеет большое число переменных, а значит, высока вероятность того, что он имеет овражную структуру.

В этом случае поиск алгоритма можно было бы проводить среди методов второго порядка. Однако существует несколько проблем. Во-первых, доказательство существования вторых производных требует наложения дополнительных ограничений на функции, определяющие правые части уравнения движения объекта. Во-вторых, реализация методов второго порядка требует эффективной оценки матрицы Гессе. Простая аппроксимация этой матрицы с помощью градиентов может оказаться очень трудоемкой. Можно попробовать получить для нее формулу, аналогичную той, что приведена в теореме из предыдущего раздела. Однако, это существенно усложнит реализацию алгоритма. В-третьих, матрица Гессе в этой задаче, вообще говоря, не является разреженной, а значит ее обращение – весьма затратная операция. С увеличением количества переменных трудоемкость алгоритма будет существенно возрастать.

С учетом приведенных проблем следует остановиться на методе сопряженных направлений ([4], [18]). Он является методом первого порядка и не требует трудоемких операций с матрицами вторых производных, при этом эффективен при оптимизации функций, имеющих овражную структуру. Реализация этого метода приведена в приложении (Листинг 3).

Глава 3. Математическая модель морского судна

Рассмотрим нелинейную модель динамики, предложенную в работе [2]. Здесь судно представляется твердым телом, движущимся в горизонтальной плоскости, что позволяет использовать его описание с помощью трех координат: положением судна на плоскости в системе координат, связанной с Землей и курсом судна, который отсчитывается от оси x .

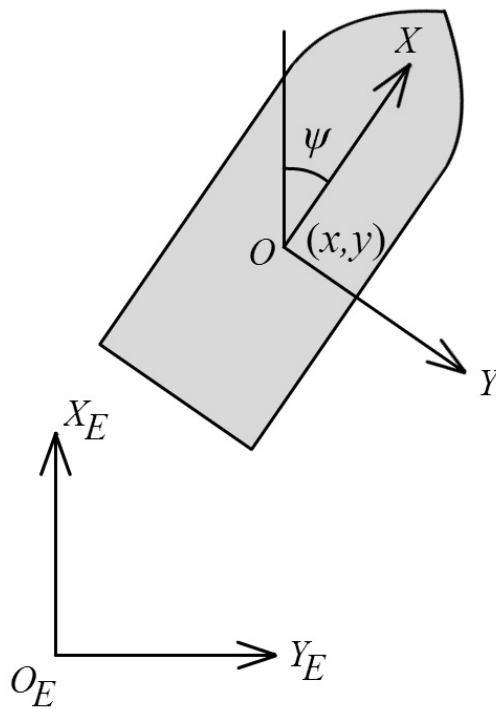


Рис.1. Расположение абсолютной и связанной систем координат.

Общее уравнение выглядит следующим образом:

$$\begin{aligned} \mathbf{M}\dot{\mathbf{v}} &= -\mathbf{D}\mathbf{v} + \boldsymbol{\tau} + \mathbf{b}, \\ \dot{\boldsymbol{\eta}} &= \mathbf{R}(\psi)\mathbf{v}. \end{aligned} \quad (11)$$

Здесь \mathbf{M} , \mathbf{D} – постоянные матрицы инерции и вязкого трения соответственно, \mathbf{v} и $\boldsymbol{\eta}$ – скорость и положение судна, $\boldsymbol{\tau}$ и \mathbf{b} – управляющие и возмущающие воздействия. Матрица \mathbf{R} имеет вид

$$\mathbf{R}(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

и описывает связь между скоростями в разных системах координат. Она является причиной нелинейности системы.

Как было сказано в постановке задачи, измеренный сигнал нуждается в предварительной обработке для фильтрации погрешностей измерения, а также оценки скорости судна и внешних возмущений. Для этого можно использовать наблюдатель, предложенный, например, в работе [2].

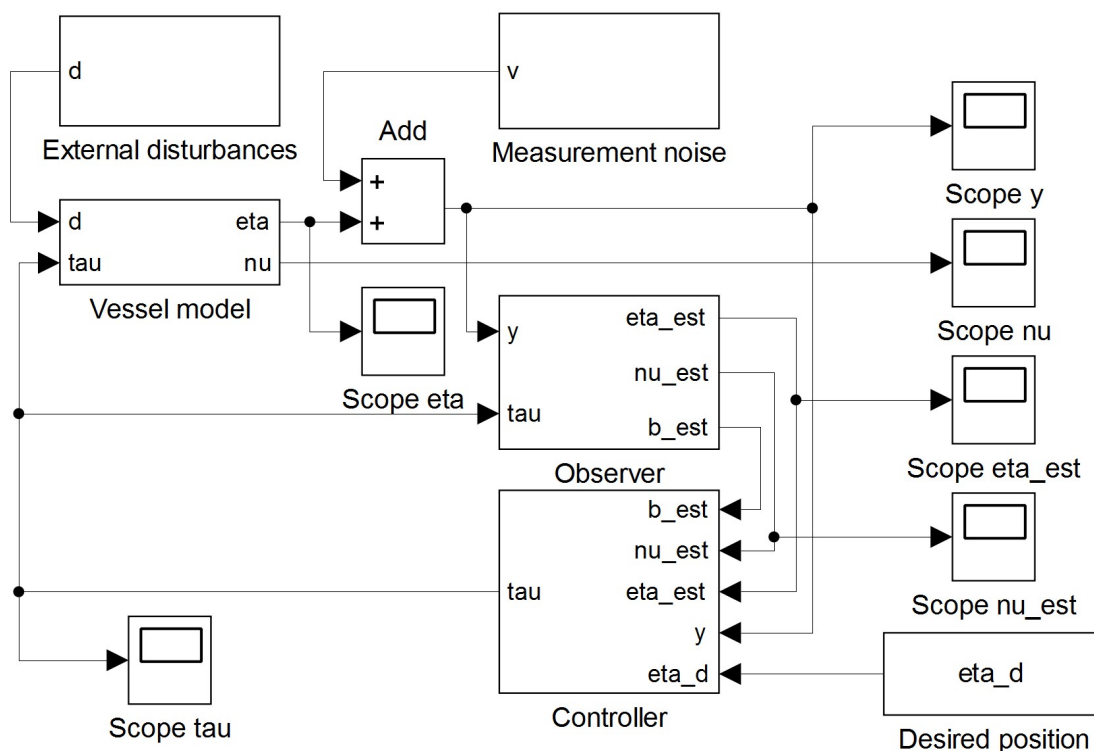
Глава 4. Имитационное моделирование движения морского судна

Компьютерное и имитационное моделирование позволяют проводить эксперименты на базе вычислительных систем без использования физических моделей объектов управления. Это помогает выявлять важные свойства процесса управления и служит опорой для выбора закона управления.

В качестве средства компьютерного моделирования динамических систем в данной работе используется среда MATLAB [7], [13], [17] с подсистемой программных средств Simulink. Этот программный комплекс позволяет строить компьютерные модели систем произвольной сложности в визуальном режиме, используя набор стандартных элементов. Система дает возможность отображать в специальных блоках данные, полученные в ходе моделирования, или передавать их в рабочую среду MATLAB для дальнейшей обработки. Этот пакет подробно представлен в книгах [14], [16] и [19].

4.1. Компьютерная модель морского судна

Компьютерная модель для проведения вычислительного эксперимента приведена на рис. 2. Она состоит из нескольких главных компонент: внешние возмущения, погрешности измерения, модель судна, закон управления, асимптотический наблюдатель. Отдельного рассмотрения требует блок закона управления. В него передаются данные о векторе состояния судна и внешних возмущений, которые оцениваются с помощью асимптотического наблюдателя. В нем эта информация передается в блок, вызывающий функцию MATLAB, в которой реализован алгоритм управления с прогнозом.



Для проведения численного эксперимента было выбрано судно “Northern Clipper”, уравнения которого даны в работе [10]. Длина судна $L = 76,2$ м, масса $m = 4,59 \cdot 10^6$, матрицы модели (11) имеют следующие значения:

Предположим, что в начальный момент времени судно расположено в начале координат $x_0 = 0 м$, $y_0 = 0 м$, $\psi_0 = 0^\circ$ и его скорость равна нулю. Для вектора желаемого положения $\boldsymbol{\eta}_d = (x_d, y_d, \psi_d)$ примем следующие значения: $x_d = 40 м$, $y_d = 40 м$, $\psi_d = 40^\circ$.

4.2. Сравнение методов вычисления градиента

Прежде всего, оценим качество работы алгоритма вычисления градиента. Сравним предложенный алгоритм с алгоритмом разностной оценки градиента функционала. Примем горизонт управления и прогноза равным 20. Произведем вычисления градиента для 10 произвольных точек. Пример результата приведен на рис. 3.

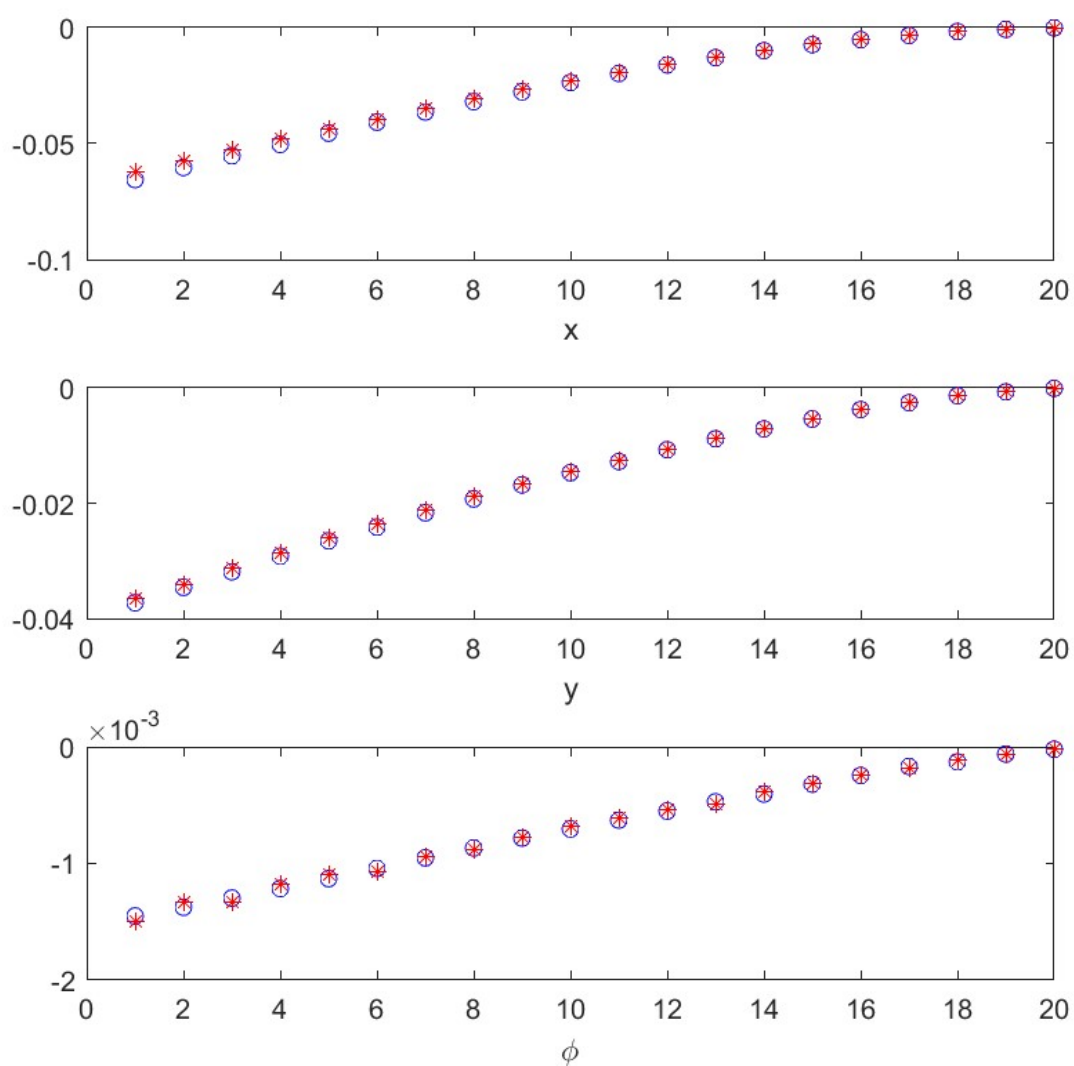


Рис. 3. Значение частных производных по переменным управления для всего горизонта прогноза.

В данном случае среднее время вычисления для метода разностной оценки градиента равняется 0.1413, в то время как предложенный метод дает результат в среднем 0.5664. То есть, для выбранных параметров, этот метод оказывается в среднем в 4 раза быстрее. Если увеличить горизонт прогноза до 40, то время составит 0.1640 и 1.2086 соответственно. Как видно в этом случае разрыв усиливается. Таким образом, алгоритм вычисления градиента работает заметно быстрее, чем простой алгоритм.

4.3. Анализ качества функционирования системы управления

Теперь перейдем к имитационному моделированию движения судна. На рис. 4 представлен переходный процесс стабилизации судна в желаемом положении.

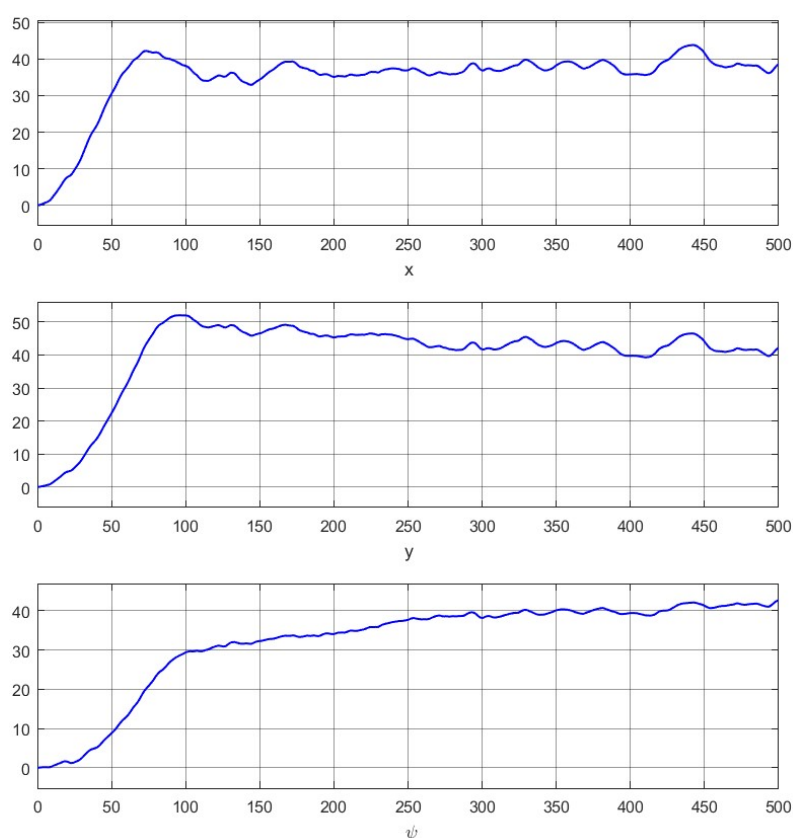


Рис. 4 Переходный процесс динамического позиционирования морского судна

Как видно, закон управления стабилизирует судно в заданном положении. Стоит отметить, что, несмотря на эффективный метод оптимизации, алгоритм тратит достаточно много времени на поиск оптимального решения. Это может быть обусловлено структурой функционала.

Рассмотрим отдельно график управляющего сигнала (рис. 5). Это кусочно-постоянная функция. Алгоритм управления ставит ограничения на управляющий сигнал, что видно на графиках.

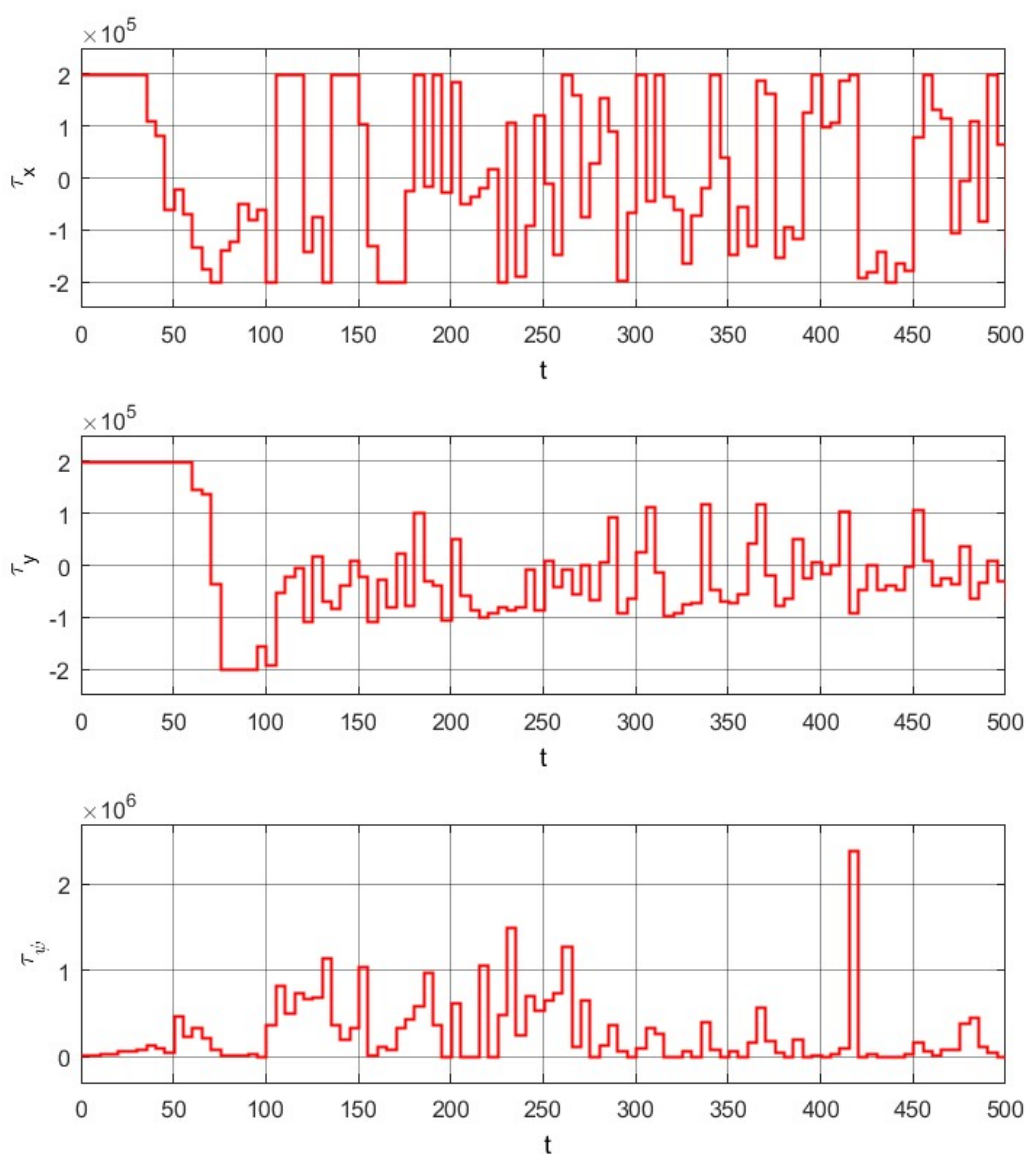


Рис. 5 Управляющий сигнал системы управления (горизонт прогноза

P=10)

Теперь попробуем увеличить горизонт прогноза. В этом случае время работы алгоритма возрастает. Но управляющий сигнал имеет меньшее количество резких перепадов (рис. 6). Это подтверждает тезис о том, что увеличение горизонта прогноза влечет улучшение качества управления.

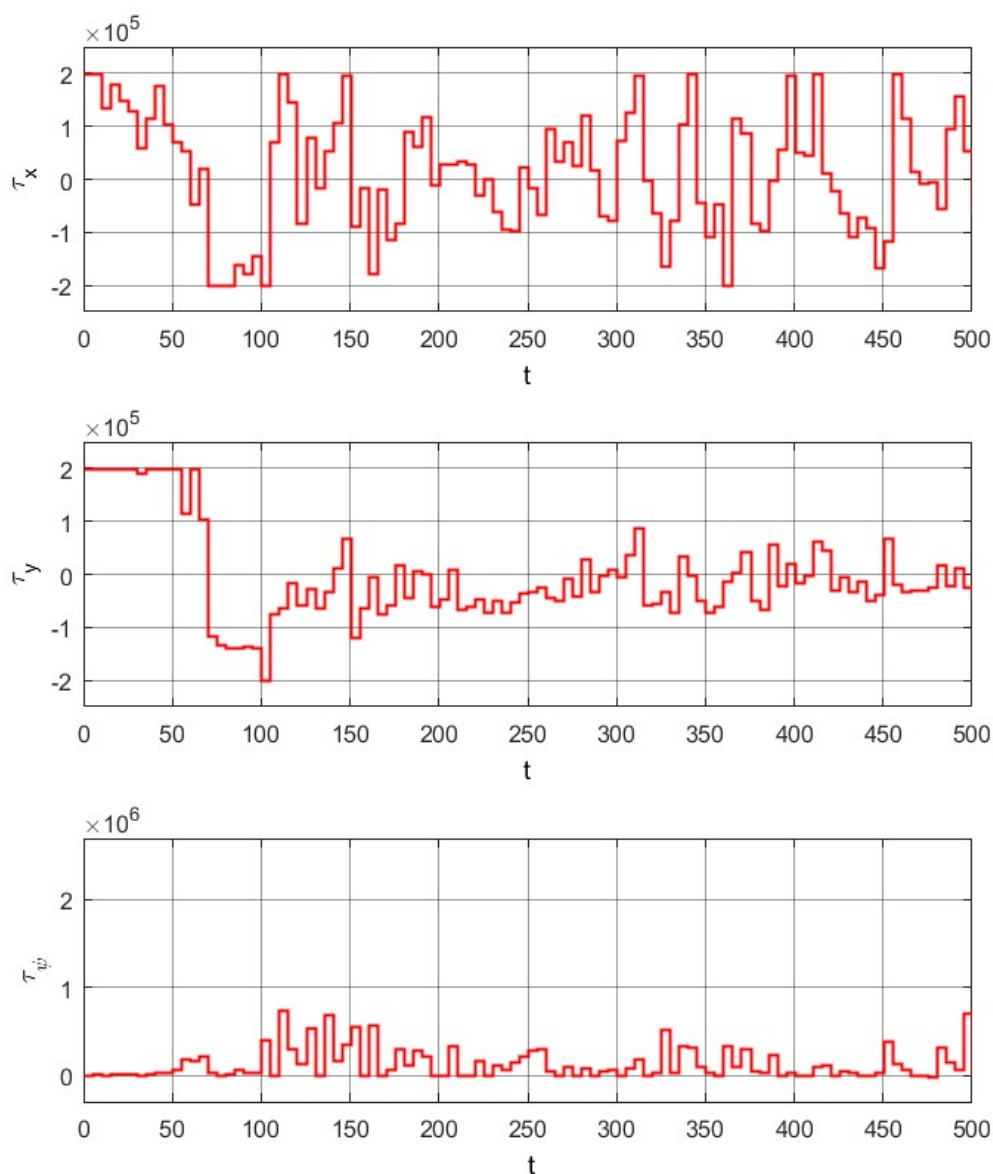


Рис. 6 Управляющий сигнал системы управления (горизонт прогноза $P=20$)

Выводы

Главная проблема на пути реализации алгоритма управления с прогнозом заключается в минимизации целевого функционала. Используя метод, рассмотренный в данной работе, можно существенно ускорить процесс вычисления градиента, а значит и сам метод оптимизации. Это достигается за счет уменьшения количества итераций для численного решения систем дифференциальных уравнений.

Как показали результаты имитационного моделирования, алгоритм управления с прогнозом стабилизирует судно в желаемом положении. Выбранный метод оптимизации дает приемлемые, с точки зрения производительности, результаты. Преимущество этого закона заключается в том, что управление, получаемое в результате работы алгоритма, является оптимальным.

Заключение

В ходе выполнения проведенных исследований получены следующие основные результаты, которые выносятся на защиту.

1. Построена математическая модель динамики судна и рассмотрен алгоритм управления с прогнозом в задаче динамического позиционирования.
2. Рассмотрен алгоритм вычисления градиента функционала качества в задаче управления с прогнозом и выбран эффективный метод оптимизации.
3. Разработан компьютерный имитационный моделирующий комплекс, в котором реализован выбранный численный метод оптимизации функционала качества.
4. Для конкретного судна, управляемого в режиме динамического позиционирования, проведен анализ алгоритма управления с прогнозом и сделаны выводы о его преимуществах и недостатках.

Список литературы

1. Balchen, J. G., N. A. Jenssen and S. Sælid. Dynamic Positioning using Kalman Filtering and Optimal Control Theory // IFAC/IFIP Symposium on Automation in Offshore Oil Field Operation, Amsterdam, The Netherlands, 1976. pp. 183—186.
2. Fossen, T. I. and J. P. Strand. Passive Nonlinear Observer Design for Ships Using Lyapunov Methods: Experimental Results with a Supply Vessel // Automatica, 1999. Vol. (35), No. (1) , pp. 3 – 16.
3. Fung, P.T.K.,&Grimble,M.J. Dynamic ship positioning using a self-tuning Kalman filter // IEEE Transactionson Automatic Control, 1983. AC-28(3), pp. 339–350.
4. Gill P. E., Murray W., Wright M.\:H. Practical Optimization. London: Academic Press, 1981. 418 p.
5. Johansen T. Toward dependable embedded model predictive control // IEEE Systems Journal. 2017. Vol. 11, No 2. P. 1208–1219.
6. Loria A., T. I. Fossen, and E. Panteley. A Separation Principle for Dynamic Positioning of Ships: Theoretical and Experimental Results // IEEE Transactions of Control Systems Technology, 2000. Vol. 8, No. 2, pp. 332–343.
7. MATLAB Getting Started Guide/ The MathWorks, Inc. – Natick, 2009. – 272 p. http://www.mathworks.com/academia/student_version/learnmatlab.pdf
8. Polak E. Computational Methods in Optimization. London: Academic Press, 1971. 329 p.
9. Strand, J. P. Nonlinear Position Control Systems Design for Marine Vessels // Doctoral Dissertation, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 1999.
10. Strand, J. P. and T. I. Fossen. Nonlinear Passive Observer for Ships with Adaptive Wave Filtering // New Directions in Nonlinear Observer Design,

Springer-Verlag London Ltd., / H. Nijmeijer and T. I. Fossen, Eds., 1999 pp. 113 – 134.

11. Sotnikova M. V., Veremey E. I. Dynamic positioning based on nonlinear MPC // 9th IFAC Conference on Control Applications in Marine Systems. 2013. Vol. 46, No 33. P. 37–42.
12. Veremey E.I. Dynamical Correction of Positioning Control Laws // Proceedings of 9th IFAC Conference on Control Applications in Marine Systems (CAMS-2103). Osaka, Japan, 2013. pp. 31–36.
13. Веремей Е. И Линейные системы с обратной связью: Учебное пособие – СПб.: Издательство «Лань», 2013. 448 с.
14. Веремей Е. И., Корчанов В.М., Коровкин М.В., Погожев С.В Компьютерное моделирование систем управления движением морских подвижных объектов. – СПб.: НИИ Химии СПбГУ, 2002. 370 с.
15. Еругин Н. П. Книга для чтения по общему курсу дифференциальных уравнений. 3-е изд. Минск: Наука и техника, 1979. 744 с.
16. Дьяконов В. П. Simulink 5/6/7: Самоучитель.– М.: ДМК-Пресс, 2008. 784 с.
17. Медведев В.С., Потёмкин В.Г. Control System Toolbox. MATLAB 5 для студентов. М., "Диалог-МИФИ", 1997, 287 с.
18. Моисеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации. М.: Наука, 1978. 352 с.
19. Черных И.В. "Simulink. Среда создания инженерных приложений" М.: Диалог-МИФИ, 2004. 496 с.

Приложение

В этом разделе приведены исходные коды программ на языке MATLAB, использовавшихся в ходе исследования.

В листинге 1 приведен код, который производит расчет значения целевого функционала.

Листинг 1.

```
function [ J,sol,Xout ] = target_func( U, x0, b )
    global Ts;
    global C;
    global P;
    global numOfCalls;
    numOfCalls = numOfCalls + 1;
    Tmax = P*Ts;
    func = genrp( U, b );
    options = odeset('RelTol',1e-1);
    sol = ode45(func,[0 Tmax],x0,options);
    J = 0;
    Xout = zeros(P,6);
    for ii = 1:P
        Xout(ii,:) = deval(sol,Ts*ii);
        J = J + one_step_cost(Xout(ii,:)', ...
            U(3*min(ii,C)+ (-2:0)));
    end
end

function [ cost ] = one_step_cost( x1, u0 )
    global eta_d Q_eta Q_tau Q_nu;
    nul = x1(1:3);
    eta1 = x1(4:6);
    cost = (eta1 - eta_d)'*Q_eta*(eta1 - eta_d) ...
        + u0'*Q_tau*u0 ...
        + nul'*Q_nu*nul;
end
```

Метод вычисления градиента функционала приведён в листинге 2.

Листинг 2.

```
function [ J, DJ ] = target_func_with_grad( U, x0, b )
    global Ts;
    global C;
    global P;
    global eta_d;
    global Q_eta;
    global Q_tau;

    [ J,sol,Xout ] = target_func( U, x0, b );

    [ FM, THM ] = fundamental_matrix(sol, b);
    DJ = zeros(1,3*C);
    DY = zeros(P,6);
    Cy = [zeros(3) eye(3)];
    X = cell(1,P);
    for ii = 1:P
        dY = Xout(ii,4:6) - eta_d';
        X{ii} = FM(Ts*ii);
        DY(ii,:) = 2*dY*Q_eta*Cy*FM(Ts*ii);
    end

    buff = cumsum(DY, 'reverse');

    DX_DU = cell(1,C);
    for ii = 1:C
        DX_DU{ii} = X{ii}\THM(Ts*ii);
    end
    for ii = C:-1:2
        DX_DU{ii} = DX_DU{ii} - DX_DU{ii-1};
    end
    for ii = 1:C
        idx = 3*ii + (-2:0);
        DJ(idx) = buff(ii,:)*DX_DU{ii} + ...
2*U(idx) '*Q_tau;
    end
end
```


Функция, реализующая метод сопряженных градиентов приведена в следующем листинге:

Листинг 3.

```
function [ xk ]=conj_grad_meth(fun, fungrad, x0, A, b)
    N = int16(length(x0));
    ii = int16(0);
    eps = 10;

    xk = x0;
    xkprev = 0;
    grad = 0;
    dk = 0;
    while true
        gradprev = grad;
        xkprev = xk;

        [ f, grad ] = fungrad( x0 );

        if ii == 0
            dk = -grad;
        else
            betta = (grad*(grad - gradprev')) ...
/(gradprev*gradprev');
            dk = -grad + betta*dk;
        end

        a = fminbnd(@(a) fun(xk + a*dk'),-10,10);
        if( a * norm(dk) < eps )
            break;
        end
        if(a < 0)
            ii = 0;
            continue;
        end

        xk = xk + a*dk'
        norm(xk - xkprev)
        ii = mod(ii + 1,N);
    end
end
```

Для измерения времени работы методов оценки градиента и вывода результатов использовался код листинга 4.

Листинг 4.

```
global C;
global tau_max;
x0 = [0;0;0;0;0;0];
b = [0;0;0];
TAU_max = zeros(3*C,1);
for ii = 1:C
    TAU_max(3*ii + (-2:0)) = tau_max;
end
simp_meth = zeros(100,1);
my_meth = zeros(100,1);
for ii = 1:100
    U0 = (2*rand(3*C,1) - 1).*TAU_max;
    J = target_func( U0, x0, b );
    DJ = zeros(3*C,1);
    tic;
    for jj = 1:3*C
        dU = zeros(3*C,1);
        eps = 100000;
        dU(jj) = eps;
        Jv = target_func( U0 + dU, x0, b );
        DJ(jj) = (Jv - J)/eps;
    end
    simp_meth(ii) = toc;

    tic;
    [ J, DJest ] = target_func_with_grad( U0, x0, b );
    my_meth(ii) = toc;

    idx = 3*(1:C);
    subplot(3,1,1)
    plot(1:C, DJest(idx-2), 'bo', 1:C, DJ(idx-2), 'r*');
    xlabel('x');
    subplot(3,1,2)
    plot(1:C, DJest(idx-1), 'bo', 1:C, DJ(idx-1), 'r*');
    xlabel('y');
    subplot(3,1,3)
    plot(1:C, DJest(idx), 'bo', 1:C, DJ(idx), 'r*');
    xlabel('\phi');
end
```